UCRL-JRNL-204137

A multigrid method for variable coefficient Maxwell's equations

J. E. Jones, B. Lee

May 13, 2004

Siam Journal of Scientific Computing

Abstract

This paper presents a multigrid method for solving variable coefficient Maxwell's equations. The novelty in this method is the use of interpolation operators that do not produce multilevel commutativity complexes that lead to multilevel exactness. Rather, the effects of multilevel exactness are built into the level equations themselves- on the finest level using a discrete $T-V$ formulation, and on the coarser grids through the Galerkin coarsening procedure of a $T-V$ formulation. These built-in structures permit the levelwise use of an effective hybrid smoother on the curl-free near-nullspace components, and these structures permit the development of interpolation operators for handling the curl-free and divergence-free error components separately, with the resulting block diagonal interpolation operator not satisfying multilevel commutativity but having good approximation properties for both of these error components. Applying operator-dependent interpolation for each of these error components leads to an effective multigrid scheme for variable coefficient Maxwell's equations, where multilevel commutativity-based methods can degrade. Numerical results are presented to verify the effectiveness of this new scheme.

Key words. Maxwell's equations, multigrid method, edge finite elements.
AMS(MOS) subject classifications. 65N55, 65N30

1 Introduction

In recent years, there has been substantial interest in solving the curl-curl formulation of Maxwell's equations in the electric or magnetic field. This formulation commonly arises in the time-domain and frequency-domain approaches for solving time-dependent electromagnetic problems. For both approaches, the challenges in developing an efficient multigrid algorithm are formidable: the time-domain leads to variable-coefficient equations with large near-nullspaces, as with the frequency-domain but with the added challenge of being indefinite. In this paper, we consider only the definite curl-curl formulation.

There exist several successful geometric ([15], [2]) and algebraic ([19], [5]) multigrid methods for solving the definite curl-curl equations. But the performance of these methods degrade as the variation of the material coefficients increases. The reason for this degradation can be traced back to the interpolation operator. For geometric schemes, even with nested finite element spaces, the natural finite element interpolation fails when the coefficients strongly vary much the same way natural finite element interpolation fails for scalar diffusion equations with rapidly varying coefficients- i.e., finite element interpolation leads to simple arithmetic averaging of the coefficients on the coarser grids. For algebraic multigrid schemes, since the interpolation operator is constructed with the constraint that a multilevel commutativity complex is formed, divergence-free error components can be untouched or even amplified by this type of interpolation. In particular, interpolation operators constructed in this manner can handle at most only the curl-free error components, and thus, only half of the Helmholtz decomposition of the error.

However, the value of a multilevel commutativity complex cannot be overlooked. This complex permits easy movement between two-term exact sequences on different grid levels. With a discretization that preserves this two-term exact sequence, nullspace components of the curl operator are explicitly known, and hence, hybrid smoothers ([15], [2]) can be constructed to effectively handle the curl-free error components. Having this two-term sequence on each level then guarantees level smoothers that eliminate curl-free error of any grid scale, and having a multilevel commutativity complex guarantees curl-free coarse grid corrections are indeed curl-free on the finer grid. But ideally, one needs a multilevel commutativity complex that spans over an exact sequence that also includes the divergence operator. This would involve developing separate interpolation operators for the curl-free and divergence-free components, which implicitly implies the existence of an exact discrete Helmholtz decomposition. Or, if an exact discrete Helmholtz decomposition does not exist, one needs an interpolation operator that acts appropriately (i.e., good approximation) on both the curl-free and divergence-free components. Such an interpolation operator is unfortunately extremely difficult to construct.

The challenge in developing a multigrid scheme for variable coefficient curl-curl equations is then constructing interpolation operators that have good approximation properties for both curl-free and divergence-free errors, and, such that the coarse grid problems constructed using these operators have the relevant effects of two-term exact sequences in order for a hybrid smoother or overlapping Schwarz smoother to be effective on coarser levels. Rather than constructing interpolation operators that satisfy these constraints, in this paper, we develop a multigrid scheme that relaxes on the multilevel commutativity constraint, and hence, has more freedom in the construction of interpolation operators to handle the curl-free and divergence-free error components. This method involves a discrete $T-V$ formulation on the finest level. The construction of this $T-V$ formulation can be obtained from the given curl-curl formulation matrix and a discrete gradient on the finest level. The goal of this formulation is to introduce the nullspace of the curl operator, i.e. gradients, explicitly into the set of equations. This leads to a discrete system for the curl-free and approximate divergence-free components of the solution, which corresponds to a system pde for these solution components. Separate operator-based interpolation operators are then constructed using the original curl-curl equations and the Laplace operator derived from the introduction of the gradients. These block diagonal interpolation operators are used in the Galerkin coarsening procedure to generate coarse grid problems for the curl-free and divergence-free components of the error. Thus, at all levels, the systems are blocked 2×2, with the diagonal blocks describing the coupling within the curl-free or divergence-free components, and with the off-diagonal blocks describing their coupling. Now equations for the curl-free, or near-nullspace gradients, are explicitly available at all levels, and so a hybrid smoother can be applied at each level.

This paper is organized as follow. In section 2, we introduce the curl-curl formulation of the definite Maxwell's equations, the functional setting for the variational problem, and the finite element spaces for the discretization. In section 3, because of their importance in understanding solution methods for the curl-curl formulation, we review the de Rham complex for the curl-curl formulation and the multilevel commutativity diagrams connecting the complexes on different levels. In particular, we examine what multilevel commutativity achieves and its significance in constructing an effective multigrid algorithm. In section 4, we describe our $T-V$ formulation and the operator-based interpolation operators to handle variable coefficients. And, in section 5, we present some preliminary numerical results illustrating the effectiveness of this multigrid scheme, and equally important, illustrating no need for multilevel commutativity complexes.

2 Curl-Curl Formulation in the Electric Field

Let $\Omega \times T$ be the Cartesian product of a bounded simply-connected domain $\Omega \in \Re^n, n = 2, 3$, and a non-negative time interval T. To guarantee appropriate regularity of the problem, let Ω have a smooth or polygonal boundary Γ. Electromagnetic phenomenom in $\Omega \times T$ can be described by the differential equations

$$\begin{aligned} \nabla \times \mathbf{E} = & \quad -\frac{\partial \mathbf{B}}{\partial t} \quad \text{in } \Omega \times T \\ \nabla \times \mathbf{H} = & \quad \frac{\partial \mathbf{D}}{\partial t} + \mathbf{j} \quad \text{in } \Omega \times T \end{aligned} \tag{1}$$

with the linear material constituitive relations

$$\begin{aligned} \mathbf{D} = & \quad \epsilon \mathbf{E} \quad \text{in } \Omega \times T \\ \mathbf{B} = & \quad \mu \mathbf{H} \quad \text{in } \Omega \times T \\ \mathbf{j} = & \quad \sigma \mathbf{E} \quad \text{in } \Omega \times T \end{aligned} \tag{2}$$

([18]). The equations in (1) are Faraday and Ampere's laws of Maxwell's equations, and the third equation in (2) is a simple Ohm's law. Here, $\mathbf{E}, \mathbf{D}, \mathbf{H}, \mathbf{B}$, and $\mathbf{j}$ are respectively the electric field, electric flux, magnetic field, magnetic flux, and electric current; and ϵ, μ, and σ are respectively the electric permittivity, magnetic permeability, and electric conductivity, which we assume all to be

spatially dependent but independent of t. Using (1) and (2), we have

$$\nabla\times\frac{1}{\mu}\nabla\times\mathbf{E}+\epsilon\frac{\partial^2\mathbf{E}}{\partial^2 t}+\sigma\frac{\partial\mathbf{E}}{\partial t}=\mathbf{0}.$$

Differencing the time derivatives, and for simplicity, assuming that the boundary surface is perfectly conducting, we obtain a boundary-value problem of the form

$$\begin{array}{rll}\nabla\times\alpha\nabla\times\mathbf{E}+\beta\mathbf{E} & =\mathbf{f} & \text{in } \Omega,\\ \mathbf{n}\times\mathbf{E} & =\mathbf{0} & \text{on } \Gamma,\end{array} \tag{3}$$

where α and β are positive functions (in $L_\infty(\Omega)$). We will call (3) the *curl-curl formulation* in the electric field. At each time-step of the solution process for the time-dependent problem, an equation of the form (3) must be solved.

To describe the variational formulation of (3), several functional spaces are needed. So, we will denote the usual k'th order Sobolev spaces by $H^k(\Omega)$ and their homogeneous trace subspaces by $H_0^k(\Omega)$. We also need spaces

$$\begin{aligned}H(\mathrm{div};\Omega) &= \{\mathbf{v}\in[L^2(\Omega)]^n:\ \nabla\cdot\mathbf{v}\in L^2(\Omega)\}\\ H(\mathrm{curl};\Omega) &= \{\mathbf{v}\in[L^2(\Omega)]^n:\ \nabla\times\mathbf{v}\in[L^2(\Omega)]^{2n-3}\}.\end{aligned}$$

We denote their homogeneous trace subspaces by $H_0(\mathrm{div};\Omega)$ and $H_0(\mathrm{curl};\Omega)$, respectively. Now, the variational formulation of (3) is to find $\mathbf{E}\in H_0(\mathrm{curl};\Omega)$ such that

$$(\alpha\nabla\times\mathbf{E},\nabla\times\mathbf{v})+(\beta\mathbf{E},\mathbf{v})=(\mathbf{f},\mathbf{v})\qquad\forall\mathbf{v}\in H_0(\mathrm{curl};\Omega). \tag{4}$$

To discretize variational problem (4), the lowest-order Nedelec finite elements are used. Let $\mathcal{T}_h := T_i$ be a quasi-uniform and shape-regular triangulation of Ω with mesh-size h ([8]). The lowest-order local Nedelec finite element space is

$$\mathcal{ND}(T_i):=\{\mathbf{v}=\mathbf{p}+\mathbf{r}:\ \mathbf{p}\in[\mathcal{P}_0(T_i)]^n;\ \mathbf{r}\in[\mathcal{P}_1(T_i)]^n \text{ with } (\mathbf{r},\mathbf{x})=0\ \forall\mathbf{x}\in T_i\},$$

where $\mathcal{P}_m(T_i)$ is the space of m'th degree polynomials in T_i. The degrees of freedom are the tangential components along the edges of T_i- along the l'th edge e_l,

$$\int_{e_l}\mathbf{t}\cdot\mathbf{v}\ ds,$$

the global finite element space is

$$\mathcal{ND}(\mathcal{T}_h):=\left\{\mathbf{v}_h\in H_0(\mathrm{curl};\Omega):\mathbf{v}_{h|T_i}\in\mathcal{ND}(T_i)\ \ \forall T_i\in\mathcal{T}_h\right\}.$$

This choice of finite element space guarantees tangential continuity of $\mathbf{v}_h$. With $\mathbf{E}_h$ denoting the discrete approximation to the solution of (4), the discrete variational problem is to find $\mathbf{E}_h\in\mathcal{ND}(\mathcal{T}_h)$ such that

$$(\alpha\nabla\times\mathbf{E}_h,\nabla\times\mathbf{v}_h)+(\beta\mathbf{E}_h,\mathbf{v}_h)=(\mathbf{f},\mathbf{v}_h)\qquad\forall\mathbf{v}_h\in\mathcal{ND}(\mathcal{T}_h). \tag{5}$$

We also will need the standard first-order scalar Lagrange finite element space:

$$\mathcal{H}_h(\mathcal{T}_h):=\left\{v\in H_0^1(\Omega):v_{|T_i}\in\mathcal{P}_1(T_i)\ \ \forall T_i\in\mathcal{T}_h\right\}$$

with the usual nodal degrees of freedom. Note that the gradient of an element of $\mathcal{H}_h(\mathcal{T}_h)$ is not only in $\mathcal{ND}_0(\mathcal{T}_h)$ but also in the nullspace of the curl operator. In fact, the assumption of simple-connectedness of Ω guarantees that the nullspace of the curl operator consists of only gradients. By cohomology theory, this is also true in the triangulated domain ([7]). (The algorithm of this paper as with other multigrid algorithms for solving (5) require simple-connectedness of Ω. For domains with holes such as practical domains with cavities, an additional procedure is needed to handle special troublesome subspaces of low dimension. These subspaces can be treated on a coarse grid ([16]).)

3 De Rham and Communativity Complexes, and Multigrid

De Rham complexes are an amazing tool for describing stable discretizations of system pde's ([1]). Their use in numerical analysis has only been realized in the past decade ([7], [17], [1]). For the Maxwell's equations, which have intrinisic topological properties reflective of de Rham complexes ([3]), these complexes have been used to explain the stablility of Nedelec finite element for the curl-curl formulation ([6], [1]). Essentially, de Rham sequences describe a relationship between the gradient, curl, and divergence operators. Given a scalar or system of partial differential equations, by *separating* the metric (e.g., material tensors or coefficients) from the topology (i.e., simple differential operators that depend only the topology of the domain), subtle geometric properties of the pde's can be exposed by the de Rham sequences. These properties should be retained in the discretization for the sake of numerical stability.

In the continuum, the complex of interest is

$$H^1(\Omega) \xrightarrow{\nabla} H(\mathrm{curl};\Omega) \xrightarrow{\nabla\times} H(\mathrm{div};\Omega) \xrightarrow{\nabla\cdot} L^2(\Omega). \tag{1}$$

Here, we have a sequence of vector spaces defined on our *contractible* domain Ω and a sequence of operators interrelating them. The vector spaces are *exact* in the sense that the codomain of an operator defined on the space to its left is in the kernel of the operator and domain space to its right- e.g., the kernel of $\nabla\times$ defined on $H(\mathrm{curl};\Omega)$ is $\nabla H^1(\Omega)$. Now let $\{H_h^1,\ H_h(\mathrm{curl}),\ H_h(\mathrm{div}),\ L_h^2\}$ and $\{H_{2h}^1,\ H_{2h}(\mathrm{curl}),\ H_{2h}(\mathrm{div}),\ L_{2h}^2\}$ be two sets of finite element spaces that need not be nested and of which the first set forms a discrete analogue to (1):

$$H_h^1(\Omega) \xrightarrow{\nabla^h} H_h(\mathrm{curl};\Omega) \xrightarrow{\nabla\times^h} H_h(\mathrm{div};\Omega) \xrightarrow{\nabla\cdot^h} L_h^2(\Omega). \tag{2}$$

Of course, this means that the finite element spaces in the first set cannot be chosen separately. In our case, choosing the lowest-order Nedelec elements for $H_h(\mathrm{curl};\Omega)$, the other finite element spaces are the first-order Lagrangian elements, the lowest-order Raviart-Thomas elements ($\mathcal{RT}_h$), and the piecewise constant elements ($\mathbf{1}_h$) for $H_h^1(\Omega)$, $H_h(\mathrm{div};\Omega)$, and $L_h^2(\Omega)$ respectively. Denoting the induced finite element interpolation operators of $\mathcal{H}_h$, $\mathcal{ND}_h$, $\mathcal{RT}_h$, and $\mathbf{1}_h$ by $\Pi_{\mathcal{H}}^h$, $\Pi_{\mathcal{ND}}^h$, $\Pi_{\mathcal{RT}}^h$, and $\Pi_{\mathbf{1}}^h$, the following commutativity complex then holds:

$$\begin{array}{llll} H^1(\Omega) & \xrightarrow{\nabla} H(\mathrm{curl};\Omega) & \xrightarrow{\nabla\times} H(\mathrm{div};\Omega) & \xrightarrow{\nabla\cdot} L^2(\Omega) \\ \downarrow \Pi_{\mathcal{H}}^h & \downarrow \Pi_{\mathcal{ND}}^h & \downarrow \Pi_{\mathcal{RT}}^h & \downarrow \Pi_{\mathbf{1}}^h \\ \mathcal{H}_h & \xrightarrow{\nabla^h} \mathcal{ND}_h & \xrightarrow{\nabla\times^h} \mathcal{RT}_h & \xrightarrow{\nabla\cdot^h} \mathbf{1}_h. \end{array} \tag{3}$$

This commutativity relates a set of continuous spaces to a set of discretizations, under a collection of differential operators. For several existing multigrid methods, additional hierarchies of commutativity relating the level spaces are needed. For example, in a two-level scheme, a multilevel commutativity complex of the form

$$\begin{array}{llll} H^1(\Omega) & \xrightarrow{\nabla} H(\mathrm{curl};\Omega) & \xrightarrow{\nabla\times} H(\mathrm{div};\Omega) & \xrightarrow{\nabla\cdot} L^2(\Omega) \\ \downarrow \Pi_{\mathcal{H}}^h & \downarrow \Pi_{\mathcal{ND}}^h & \downarrow \Pi_{\mathcal{RT}}^h & \downarrow \Pi_{\mathbf{1}}^h \\ \mathcal{H}_h & \xrightarrow{\nabla^h} \mathcal{ND}_h & \xrightarrow{\nabla\times^h} \mathcal{RT}_h & \xrightarrow{\nabla\cdot^h} \mathbf{1}_h \\ \uparrow P_{H^1}^h & \uparrow P_{H(\mathrm{curl})}^h & \uparrow P_{H(\mathrm{div})}^h & \uparrow P_{L^2}^h \\ H_{2h}^1 & \xrightarrow{\nabla^{2h}} H_{2h}(\mathrm{curl}) & \xrightarrow{\nabla\times^{2h}} H_{2h}(\mathrm{div}) & \xrightarrow{\nabla\cdot^{2h}} L_{2h}^2, \end{array} \tag{4}$$

where $P_{(\cdot)}^h$ are multilevel interpolation operators that are constructed to satisfy commutativity, may be desired.

It is insightful to examine what is actually gained in constructing multilevel interpolation that satisfies commutativity. Consider a multigrid scheme for solving (3) that has commutativity interpolation. This scheme will involve only the two left columns of (4). In this scheme, coarse scale

near-nullspace gradient components of (3) are interpolated to fine scale near-nullspace gradient components. In particular, when $\{H^1_{2h},\ H_{2h}(\mathrm{curl})\}$ is a nested subspace of $\{\mathcal{H}_h,\ \mathcal{ND}_h\}$ and the $P_{(\cdot)}$'s are the natural finite element interpolation operators, by the our assumption of quasi-uniformity, finite element approximation theory implies multilevel approximation property for the algebraically smooth eigenfunctions that are gradients. For algebraic multigrid on non-nested finite elements, multilevel approximation property for the algebraically smooth gradient eigenfunctions also holds. Physically, this seems strange since the coarse degrees of freedom in algebraic multigrid are located on coarse edges that are generally not a subset of fine edges- i.e., physically, electric fields that have zero circulation over a given closed cycle (e.g., coarse cycle) generally do not have zero circulation over closed cycles not contained in that given cycle (e.g., fine cycles). But this "contradiction" can be explained away with the commutativity relation

$$\nabla^h P^h_{H^1} = P^h_{H(\mathrm{curl})} \nabla^{2h}, \tag{5}$$

i.e., a field with zero circulation on a coarse cycle corresponds to a coarse gradient, whose generating potential can be interpolated to a fine scalar function and then operated on with the fine gradient operator, to produce a zero circulation field over the fine cycles. Hence, non-nesting of the coarse edges *does not pose* a problem. We revisit this later.

Probably the greatest gain in using commutativity interpolation is possible computational reduction. With such interpolation operators, we can ascend up any one column of (4) using only this column's associated interpolation and degrees of freedom. For example, for Maxwell problems that have *only* curl-free near nullspaces, having interpolation operators that satisfy (5) means that only the action of $P^h_{H(\mathrm{curl})}$ and only coarse edge degrees of freedom are required. $P^h_{H^1}$ and nodal degrees of freedom are not explicitly used in the multigrid solve phase. Given an arbitrary coarse edge function that has a curl-free and a divergence-free component, these operators interpolate the curl-free component correctly. As explained above, this coarse zero circulation component is interpolated onto fine cycles by implicitly taking the long route $\nabla^h P^h_{H^1}$.

Moreover, commutativity interpolation produces exact sequence structures on the coarser levels. Having this structure on a coarse level then leads to simple relaxation procedures that can handle the near-nullspace components of the curl and divergence operators on this coarse level. Although this is a major component in an efficient multigrid algorithm, it may not be enough. We contend that multilevel discrete de Rham structures are needed. To explain this, one needs to return to the topological basis of a de Rham complex and its discrete analogue:

De Rham complexes arise in the development of a calculus on a smooth n-dimensional manifold $\mathcal{M}$. On $\mathcal{M}$ are defined linear spaces $\mathcal{F}^p(\mathcal{M})$ of differential p-forms, i.e. alternating p-linear maps on the tangent space of $\mathcal{M}$ ([4], [13]). The spaces of p-forms are connected through an exterior differential operator $d : \mathcal{F}^p(\mathcal{M}) \to \mathcal{F}^{p+1}(\mathcal{M})$, which satisfies Stokes' theorem over any sub-manifold of $\mathcal{M}$:

$$\int_\Sigma d\omega = \int_{\partial\Sigma} \omega \qquad \forall\ \omega \in \mathcal{F}^p(\mathcal{M}),\ \Sigma \subset \mathcal{M}, \tag{6}$$

where Σ is a smooth and oriented $(p+1)-$dimensional sub-manifold of $\mathcal{M}$ and $\partial\Sigma$ is its $p-$dimensional boundary. The connection through d forms a de Rham complex:

$$\mathcal{F}^0 \xrightarrow{d} \mathcal{F}^1 \xrightarrow{d} \cdots \xrightarrow{d} \mathcal{F}^n. \tag{7}$$

Exactness $dd = 0$ follows from the topological relation $\partial\partial\Sigma = 0$ ([13]) and Stokes' theorem, demonstrating a strong connection between exactness and boundaries of sub-manifolds. Now, to create a discrete analogue of this complex, one must parametrize $\mathcal{M}$ into sub-manifolds, and define a finite-dimensional space to approximate $\mathcal{F}^p$ on the sub-manifolds. Parametrization of $\mathcal{M}$ is obtained through p-cubes or Euclidean simplices: 0-cubes, 1-cubes, $\cdots$, n-cubes. *Chains* of p-cubes then allow the construction of general p-dimensional sub-manifolds $\mathcal{M}_p$. On $p-$chains are defined mappings $C^p(\mathcal{M}_p) : \mathcal{M}_p \to \Re^N$, where N is the number of basis chains forming $\mathcal{M}_p$. These mappings are

called cochains, and from the (6), a discrete exterior derivative $d_h : C^p(\mathcal{M}_p) \to C^{p+1}(\mathcal{M}_{p+1})$ is defined. Finally, on $C^p(\mathcal{M}_p)$ are defined bijective linear mappings that take it into a space of differential p-forms. These mappings are rigorously defined so that the finite-dimensional (by bijectivity) range space of p-forms are simple, unisolvent, conforming, satisfy exactness, etc. ([14]).

When $\mathcal{M}$ is a domain in $\Re^3$, complex (7) can be identified with complex (1), 0-cubes, 1-cubes, 2-cubes, and 3-cubes are respectively points, edges, faces, and volumes, and the space of discrete differential p-forms can be taken to be $\{\mathcal{H}_h,\ \mathcal{ND}_h,\ \mathcal{RT}_h,\ \mathbf{1}_h\}$. The location of degrees of freedom of this set of finite element spaces indeed reflects a *consistent* choice of simplices to guarantee exactness. For multigrid, to guarantee multilevel exactness, a consistent choice of coarse simplices is additionally needed. In [19] and [5], this consistent choice is obtained by coarsening the edges using an "auxiliary" nodal coarsening procedure. This consistent choice is also illustrated in our previous circulation free example. However, when the coarse edges are non-nested, the set of *weighted* finite element spaces $\{H^1_{2h},\ H_{2h}(\mathrm{curl})\}$ generated by $P^h_{H(\mathrm{curl})}$ is not the proper p-forms for these coarse edges. Hence, a multilevel de Rham structure is not obtained. A multilevel de Rham structure can be obtained by choosing the coarse edges to be chains of fine edges, as is accomplished in geometric multigrid. But the implicit choice of coarse nodes here can result in low-order interpolation.

So, what does a multilevel de Rham structure give? Suppose the set of coarse finite elements is $\{\mathcal{H}_{2h},\ \mathcal{ND}_{2h},\ \mathcal{RT}_{2h},\ \mathbf{1}_{2h}\}$ on the coarse simplices, which form a quasi-uniform triangulation but are not necessarily nested with the fine simplices. Rather than stacking these complexes into three layers as in (4), we consider two complexes, each relating a continuous de Rham sequence to a discrete de Rham sequence:

$$\begin{array}{ccccccc} H^1(\Omega) & \xrightarrow{\nabla} & H(\mathrm{curl};\Omega) & \xrightarrow{\nabla\times} & H(\mathrm{div};\Omega) & \xrightarrow{\nabla\cdot} & L^2(\Omega) \\ \downarrow \Pi^{2^i h}_{\mathcal{H}} & & \downarrow \Pi^{2^i h}_{\mathcal{ND}} & & \downarrow \Pi^{2^i h}_{\mathcal{RT}} & & \downarrow \Pi^{2^i h}_{\mathbf{1}} \qquad i = 0, 1. \\ \mathcal{H}_{2^i h} & \xrightarrow{\nabla^{2^i h}} & \mathcal{ND}_{2^i h} & \xrightarrow{\nabla\times^{2^i h}} & \mathcal{RT}_{2^i h} & \xrightarrow{\nabla\cdot^{2^i h}} & \mathbf{1}_{2^i h} \end{array} \tag{8}$$

The set of coarse finite elements are stable, and by the finite element approximation property for each of these spaces, we have good multigrid approximation property everywhere. For example, for Maxwell's equations,

$$\begin{aligned} \|\mathbf{v}_h - \mathbf{v}_{2h}\|_{\mathrm{curl}} &\leq \|\mathbf{v} - \mathbf{v}_h\|_{\mathrm{curl}} + \|\mathbf{v} - \mathbf{v}_{2h}\|_{\mathrm{curl}} \\ &\leq Ch \end{aligned}$$

for all sufficiently smooth functions $\mathbf{v}$, whereas when a multilevel de Rham structure does not hold

$$\begin{aligned} \|\mathbf{v}_h - P^h_{\mathrm{curl}}\mathbf{v}_{2h}\|_{\mathrm{curl}} &\leq \|\mathbf{v} - \mathbf{v}_h\|_{\mathrm{curl}} + \|\mathbf{v} - P^h_{\mathrm{curl}}\mathbf{v}_{2h}\|_{\mathrm{curl}} \\ &\leq Ch + \|\mathbf{v} - P^h_{\mathrm{curl}}\mathbf{v}_{2h}\|_{\mathrm{curl}}, \end{aligned}$$

which is not necessarily Ch (e.g., $\mathbf{v}_h$ is divergence-free). This does not present a problem when only curl-free vectors are in the near-nullspace, but it will when α is variable.

We conclude this section by collecting our observations on commutativity interpolation. This type of interpolation gives multilevel exactness, which translates to essentially having an explicit Helmholtz decomposition of the error on each coarse level. Hence, the relaxation component of an efficient multigrid algorithm is available. This type of interpolation also gives good coarse grid approximation to the near-nullspace gradient vectors, and hence, the coarse grid correction component of an efficient multigrid algorithm for *these* vectors. Finally, commutativity interpolation leads to computer memory and computational reduction in the solve phase of a multigrid process. However, for an efficient multigrid scheme, one needs only some explicit type of Helmholtz decomposition on each level, and one clearly needs interpolation operators that give good coarse grid approximation to each part of this decomposition. To achieve both of these, explicitly having nodal (to represent

the curl-free vector) and edge (to represent the divergence-free vector) degrees of freedom and separate interpolation operators for each part of the Helmholtz decomposition are desirable. This will require more computer memory and additional computational cost per multigrid cycle, but will give robustness and simplicity.

4 $T-V$ Formulation

As one can derive from the last section, the overall advantage of commutativity interpolation is reduction in computational cost and memory in the solve phase, but at the expense of complicated construction and poor approximation to the divergence-free error components. Moreover, the computational reduction is achieved only in the solve phase- the formation of the auxiliary nodal interpolation and coarse grid operators are explicitly done in the setup phase but then discarded afterwards ([19], [5]). We would like to develop an algorithm that recycles this discarded nodal information in the solve phase. Of course, this will lead to actual nodal structures in the solve phase. But, by keeping these structures, interpolation operators can be easily constructed without the commutativity constraints, and be constructed such that the coarse grid approximation property holds for all required bad components. The motivation behind such an approach is to develop an algorithm that can be easily fitted into the framework of existing multigrid codes.

To accomplish our goal, consider curl-curl equation (4) with the solution decomposition

$$\mathbf{E} = \mathbf{E}_1 + \nabla\phi, \qquad \mathbf{E}_1 \in H_0(\mathrm{curl};\Omega),\ \phi \in H_0^1(\Omega),$$

and the variational problem: find $(\mathbf{E}_1, \phi) \in H_0(\mathrm{curl};\Omega) \times H_0^1(\Omega)$ such that

$$(\alpha\nabla\times(\mathbf{E}_1+\nabla\phi), \nabla\times(\mathbf{v}+\nabla\psi)) + (\beta(\mathbf{E}_1+\nabla\phi), (\mathbf{v}+\nabla\psi)) = (\mathbf{f}, (\mathbf{v}+\nabla\psi)) \tag{1}$$

for all $\mathbf{v} \in H_0(\mathrm{curl};\Omega),\ \psi \in H_0^1(\Omega)$. Respectively denoting the basis functions of $H_0(\mathrm{curl};\Omega)$ and $H_0^1(\Omega)$ as $\{\mathbf{v}_l\}$ and $\{\psi_i\}$, (1) can be written as a system variational problem

$$\begin{bmatrix} A_{ee} & A_{en} \\ A_{ne} & A_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{E}_1 \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{f}_e \\ \mathbf{f}_n \end{bmatrix}, \tag{2}$$

where

$$A_{ee} = [(\alpha\nabla\times\mathbf{v}_l, \nabla\times\mathbf{v}_m) + (\beta\mathbf{v}_l, \mathbf{v}_m)] \quad A_{en} = [(\beta\mathbf{v}_l, \nabla\psi_j)]$$
$$A_{ne} = [(\beta\nabla\psi_i, \mathbf{v}_m)] \quad A_{nn} = [(\beta\nabla\psi_i, \nabla\psi_j)]$$

and

$$\begin{bmatrix} \mathbf{f}_e \\ \mathbf{f}_n \end{bmatrix} = \begin{bmatrix} (\mathbf{f}, \mathbf{v}_l) \\ (\mathbf{f}, \nabla\psi_i), \end{bmatrix}$$

and for all $(\mathbf{v}_l, 0)^t, (\mathbf{0}, \psi_i)^t \in H_0(\mathrm{curl};\Omega)\times H_0^1(\Omega)$. To examine the near-nullspace component $\left(\tilde{\mathbf{E}}, \tilde{\phi}\right)^t$ of (2), let $\tilde{\mathbf{E}}$ have the Helmholtz decomposition

$$\tilde{\mathbf{E}} = \tilde{\mathbf{E}}_d + \tilde{\mathbf{E}}_c,$$

where $\tilde{\mathbf{E}}_d$ is divergence-free and $\tilde{\mathbf{E}}_c$ is curl-free. For near-nullspace component with curl-free $\tilde{\mathbf{E}}$, $\tilde{\mathbf{E}}_c = \nabla\eta$, let us determine the corresponding nodal component $\tilde{\phi}$. From the second equation of (2), the nodal component must satisfy

$$(\beta\nabla\tilde{\phi}, \nabla\psi_i) \approx -(\beta\nabla\psi_i, \tilde{\mathbf{E}}_c) = -(\beta\nabla\psi_i, \nabla\eta) \qquad \forall i.$$

Thus $\tilde{\phi}$ is approximately $-\eta$. Substituting this into the first equation, we have

$$(\alpha\nabla\times\tilde{\mathbf{E}}_c, \nabla\times\mathbf{v}_l) + (\beta\tilde{\mathbf{E}}_c, \mathbf{v}_l) - (\beta\mathbf{v}_l, \nabla\eta) = (\beta\nabla\eta, \mathbf{v}_l) - (\beta\mathbf{v}_l, \nabla\eta) = 0$$

so that $(\tilde{\mathbf{E}}, \tilde{\phi})^t = (\nabla\eta, -\eta)^t$ is indeed a near-nullspace component. Next, consider the near-nullspace component with divergence-free $\tilde{\mathbf{E}}$, $\tilde{\mathbf{E}} = \tilde{\mathbf{E}}_d$. We will assume that α, β are sufficiently smooth and consider only interior estimates so that boundary terms produced by integration by parts can be omitted (near-nullspace components are locally interior). From the second equation of (2) and integration by parts,

$$\begin{aligned} -(\nabla\cdot\beta\tilde{\mathbf{E}}_d, \psi_i) + (\beta\nabla\tilde{\phi}, \nabla\psi_i) &= -(\nabla\beta\cdot\tilde{\mathbf{E}}_d + \beta\nabla\cdot\tilde{\mathbf{E}}_d, \psi_i) + (\beta\nabla\tilde{\phi}, \nabla\psi_i) \\ &= -(\nabla\beta\cdot\tilde{\mathbf{E}}_d, \psi_i) + (\beta\nabla\tilde{\phi}, \nabla\psi_i) \approx 0 \qquad \forall i, \end{aligned} \tag{3}$$

or

$$(\beta\nabla\tilde{\phi}, \nabla\psi_i) \approx (\nabla\beta\cdot\tilde{\mathbf{E}}_d, \psi_i) \qquad \forall i.$$

If we further assume that β is smooth in the sense that $\|\nabla\beta\| \leq \epsilon$, then using elliptic regularity and the Cauchy-Schwarz inequality,

$$\|\tilde{\phi}\|_1 \leq C\|\nabla\beta\cdot\tilde{\mathbf{E}}_d\| \leq C\|\nabla\beta\|\|\tilde{\mathbf{E}}_d\| \leq C\epsilon\|\tilde{\mathbf{E}}_d\|.$$

That is, $\tilde{\phi}$ has small $H1$ energy, or equivalently, is a smooth eigenfunction of the diffusion operator $[\nabla\cdot\beta\nabla]$. Now for the first equation of (2). We have

$$(\alpha\nabla\times\tilde{\mathbf{E}}_d, \nabla\times\mathbf{v}_l) + (\beta\tilde{\mathbf{E}}_d, \mathbf{v}_l) \approx -(\beta\nabla\tilde{\phi}, \mathbf{v}_l) \qquad \forall l, \tag{4}$$

and because $\tilde{\phi}$ has small $H1$ energy, $\tilde{\mathbf{E}}_d$ has small scaled $H(\text{curl})$ energy in the order of the size of the eigenvalue corresponding to $\tilde{\phi}-$ i.e., coercivity of the curl-curl bilinear form implies

$$\|\tilde{\mathbf{E}}_d\|_{H(\text{curl})} \leq C\|\beta\nabla\tilde{\phi}\|$$

and $\tilde{\phi}$ being a smooth orthnormal eigenfuction implies

$$\|\tilde{\phi}\|_1 \leq \|\tilde{\phi}\|_2 \leq C\lambda.$$

So, $\tilde{\mathbf{E}}_d$ is a smooth eigenfunction of the operator $[\nabla\times\alpha\nabla\times + \beta I]$. Moreover, using the identity

$$\begin{aligned} \nabla\times\alpha\nabla\times\mathbf{v} &= \nabla\alpha\times(\nabla\times\mathbf{v}) + \alpha\nabla\times\nabla\times\mathbf{v} \\ &= \nabla\alpha\times(\nabla\times\mathbf{v}) - \alpha\boldsymbol{\Delta}\mathbf{v} + \nabla\nabla\cdot\mathbf{v}, \end{aligned}$$

$\tilde{\mathbf{E}}_d$ is a smooth eigenfunction of

$$[-\alpha\boldsymbol{\Delta} + \nabla\alpha\times\nabla\times + \beta I]. \tag{5}$$

Thus, the near-nullspace components of (2) are

$$\begin{pmatrix} \tilde{\mathbf{E}} \\ \tilde{\phi} \end{pmatrix} = \begin{pmatrix} \nabla\eta \\ -\eta \end{pmatrix} \qquad \text{and} \qquad \begin{pmatrix} \tilde{\mathbf{E}} \\ \tilde{\phi} \end{pmatrix} = \begin{pmatrix} \text{small scaled H(curl) norm} \\ \text{small scaled H1 norm} \end{pmatrix}$$

Multigrid interpolation operators must be designed to approximate these components well.

Note that the $\tilde{\mathbf{E}}$ component of the system near-nullspace is the near-nullspace of (4). These must be well approximated by $P^h_{H(\text{curl})}$. Note also that when α, β are constants, the divergence-free near-nullspace components are smooth eigenfunctions of $[-\alpha\boldsymbol{\Delta} + \beta I]$. Hence, to obtain good approximation to these near-nullspace components, $P^h_{H(\text{curl})}$ should act like linear interpolation on the edge degrees of freedom. Constructing $P^h_{H(\text{curl})}$ under the commutativity constraints using $P^h_{H^1}$

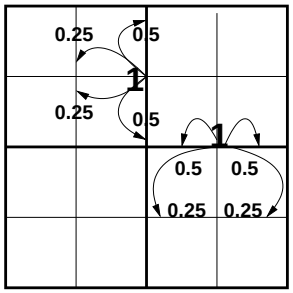

Figure 1: Nedelec linear interpolation of horizontal and vertical edges.

generally will not attain this property. A special case when this is attained occurs when the edges are nested and $P^h_{H(\mathrm{curl})}$ is the Nedelec finite element interpolation operator (see Fig. 1).

Having determined the qualitative form of the near-nullspace of (2), we now develop a multigrid algorithm. First, system (2) is discretized over $\mathcal{ND}(\mathcal{T}_h)\times\mathcal{H}_h(\mathcal{T}_h)$, which approximates the continuous near-nullspace well ([6]):

$$\begin{bmatrix} A^h_{ee} & A^h_{en} \\ A^h_{ne} & A^h_{nn} \end{bmatrix} \begin{bmatrix} \mathbf{E}^h_1 \\ \psi^h \end{bmatrix} = \begin{bmatrix} \mathbf{f}^h_e \\ \mathbf{f}^h_n \end{bmatrix}. \tag{6}$$

From the above discussion, for the curl-free near nullspace, since $\eta = -\tilde{\phi}$, we need only to approximate the diffusion operator $[-\nabla\cdot\beta\nabla]$ on the coarser grids. Good approximation to this same operator is also required in handling the nodal $\tilde{\phi}$ component of the divergence-free near-nullspace. For the edge component of this divergence-free vector, good approximation to vector "convection"-diffusion operator (5) or the curl-curl operator itself is needed on the coarser levels. To accomplish both of these approximation requirements, separate interpolation operators are used for the nodal and edge degrees of freedom. The nodal interpolation P_n will be based on the diffusion operator and the edge interpolation P_e will be based on the curl-curl operator. P_n can be constructed using AMG or box-mg techniques ([10]), and P_e can be constructed using AMGe ([9]) or AMG techniques, since the convection-diffusion operator is being implicitly coarsened. The overall interpolation operator for (6) is

$$\begin{bmatrix} P_e & 0 \\ 0 & P_n \end{bmatrix}.$$

The validity of this block-diagonal interpolation operator was based under the assumption that $\|\nabla\beta\|$ is small. When $\|\nabla\beta\|$ is large, a full 2×2 interpolation operator may be needed (see [11], [12] for a numerical comparison of diagonal block and full block interpolation operators on system pde's).

With block-diagonal interpolation, the coarse grid operator is formed using Galerkin coarsening:

$$\begin{aligned} \begin{bmatrix} A^{2h}_{ee} & A^{2h}_{en} \\ A^{2h}_{ne} & A^{2h}_{nn} \end{bmatrix} &= \begin{bmatrix} P_e & 0 \\ 0 & P_n \end{bmatrix}^t \begin{bmatrix} A^h_{ee} & A^h_{en} \\ A^h_{ne} & A^h_{nn} \end{bmatrix} \begin{bmatrix} P_e & 0 \\ 0 & P_n \end{bmatrix} \\ &= \begin{bmatrix} P_e^t A^h_{ee} P_e & P_e^t A^h_{en} P_n \\ P_n^t A^h_{ne} P_e & P_n^t A^h_{nn} P_n \end{bmatrix}. \end{aligned} \tag{7}$$

Applying this recursively, the coarse grid operator on level i can be constructed.

This algorithm requires the additional computation and storage of the off-diagonal blocks $A^{2^ih}_{en} = (A^{2^ih}_{ne})^t$. However, as the above discussion suggests, the coarse grid problems capture the divergence-free and curl-free components separately- i.e., respectively using $A^{2^ih}_{ee}$ and $A^{2^ih}_{nn}$ only. Hence, omitting

the off-diagonal blocks can also lead to an effective scheme. With the off-diagonal blocks, curl-free error components re-introduced when solving the edge correction can be surpressed.

This algorithm also appears to require more user data than just the fine grid stiffness matrix A^h and a fine grid discrete gradient operator G^h_{en} : nodes $\rightarrow$ edges, as required by commutativity-based multigrid methods. But, as in the commutativity-based methods, A^h_{nn} can be gotten using sparse matrix products and $A^h_{en} = (A^h_{ne})^t$ can be gotten freely from this same matrix product:

$$\begin{aligned} A^h_{nn} &= (G^h_{en})^t \left[A^h_{ee} G^h_{en} \right] \\ A^h_{en} &= A^h_{ee} G^h_{en}. \end{aligned}$$

Note that once these operators are formed, discrete gradient operators are not needed on the coarser levels. This apparent minor detail actually implicates some structural differences between this multigrid algorithm and commutativity-based ones. Because these operators are not needed on the coarser grids, there need not be a connection between the coarse nodal and edge grids, although the nodal and edge degrees of freedom are connected through the off-diagonal blocks of (7). This implies that the coarse simplices do not have to satisfy a consistency condition- i.e., nodes and edges can be coarsened separately. In turn, this means that a two-term exactness property does not hold on the coarser levels. Even if coarse simplices were consistency chosen, an exactness property may still not hold because the weights generated by the separate interpolation operators lead to discrete differential operators that may not be consistent. All of this should not be surprising because we independently constructed the edge interpolation to capture mainly the divergence-free error and the nodal interpolation to capture the curl-free error, rather than having one edge interpolation constructed to produce exactness to capture the curl-free error. There also should be no concern that the exactness property does not hold on coarser levels because the nodal coarse grid problem defines the equation for the curl-free error- i.e., we have a relaxation procedure for these errors on the coarser levels. With no exactness property on the coarser levels, the multilevel complex for this algorithm is

$$\begin{array}{ccc} H^1(\Omega) & \xrightarrow{\nabla} & H(\mathrm{curl};\Omega) \\ \downarrow \Pi^h_{\mathcal{H}} & & \downarrow \Pi^h_{\mathcal{ND}} \\ \mathcal{H}_h & \xrightarrow{\nabla^h} & \mathcal{ND}_h \\ \uparrow P^h_n & & \uparrow P^h_e \\ H^1_{2h} & \overset{A^{2h}_{en}}{\cdots} & H_{2h}(\mathrm{curl}) \\ \vdots & & \vdots \\ H^1_{2^i h} & \overset{A^{2^i h}_{en}}{\cdots} & H_{2^i h}(\mathrm{curl}), \end{array}$$

where $H^1_{2^i h}$ and $H_{2^i h}(\mathrm{curl})$ are subspaces of $\mathcal{H}_h$ and $\mathcal{ND}_h$, and $\overset{A^{2^i h}_{en}}{\cdots}$ denotes only a connection between the node and edge degrees of freedom, not an exactness relation.

4.1 Interpolation Schemes

There are many well-developed interpolation schemes for variable-coefficient problems. In this paper, we will consider only structured grids, and hence, employ a box-mg scheme ([10]) for P_n and an element agglomeration scheme ([9]) for P_e.

The box-mg technique is well known. Let the 9-point stencil of A^h_{nn} be

$$\left[\begin{array}{ccc} a^{nw}_{nn} & a^{n}_{nn} & a^{ne}_{nn} \\ a^{w}_{nn} & a^{c}_{nn} & a^{e}_{nn} \\ a^{sw}_{nn} & a^{s}_{nn} & a^{se}_{nn} \end{array} \right]_{IJ}$$

at coarse node IJ. This technique constructs P_n to satisfy

$$A^h_{nn} P_n \phi^{2h} = 0. \tag{1}$$

For fine nodes that underly a coarse node, the interpolation weight is one; for fine nodes that are horizontally or vertically adjacent to a coarse node, stencil collapsing is used to determine the weights- e.g. for fine nodes that are directly east or west of node IJ, the interpolation weights are respectively

$$P^e_n = -\frac{(a^{ne}_{nn} + a^e_{nn} + a^{se}_{nn})_{IJ}}{(a^n_{nn} + a^c_{nn} + a^s_{nn})_{IJ}} \qquad P^w_n = -\frac{(a^{nw}_{nn} + a^w_{nn} + a^{sw}_{nn})_{IJ}}{(a^n_{nn} + a^c_{nn} + a^s_{nn})_{IJ}}.$$

The weights for all other fine nodes are obtained using these computed weights and relation (1).

To describe the AMGe technique to construct P_e, we will refer to the two-dimensional uniform agglomerate shown in Fig. (2). We divide the fine edges of this agglomerate into boundary (b) and

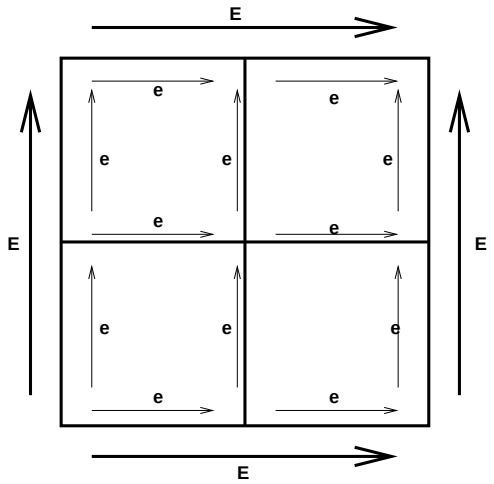

Figure 2: Coarse cell agglomerate. Coarse edges are labeled with E and fine edges are labeled with e.

interior (i) edges, and order them so that locally A^h_{ee} and P_e can be written as

$$A^h_{ee} = \begin{bmatrix} A^{bb}_{ee} & A^{bi}_{ee} \\ A^{ib}_{ee} & A^{ii}_{ee} \end{bmatrix} \qquad P_e = \begin{bmatrix} P^b_e \\ P^i_e \end{bmatrix}.$$

The boundary edges are the fine edges e that nest a coarse edge E, and the interior edges are the remaining fine edges of the agglomerate. For the boundary edges, the interpolation weights can be defined physically. Since the edge degrees of freedom are voltage drops along the edges, the boundary weights are simply the ratio of the length of the boundary edge to the length of coarse edge that it nests. With these boundary weights, the interior weights are given by the local inversion

$$P^i_e = -[A^{ii}_{ee}]^{-1} A^{ib}_{ee} P^b_e.$$

4.2 Relaxation Schemes

Since the equations for both the nodes and edges are Poisson-like, pointwise Gauss-Seidel can be used for each degree of freedom. But as in system pde's, the sweeping order of relaxation produces different smoothers. One can relax all the nodes of the nodal grid and then all the edges of the edge grid, to give the hybrid block Gauss-Seidel scheme of [15]. One can also apply a multiplicative Schwarz smoother similar to the one of [2], but where now a node and all the edges branching out of it

are solved simultanuously. This is more costly than the pointwise smoother, but it gives robustness. Yet an intermediate ordering is to update a node and then pointwise update all the edges branching out of this node in turn. Since an edge connects two nodes, a given edge is updated more than once in a sweep. Hence, this smoother is more costly than the pointwise hybrid smoother.

5 Numerical Experiments

This multigrid algorithm has currently been tested on two-dimensional uniform grids. Larger problems, three-dimensional problems, and algebraic multigrid variants of this algorithm will be presented in a forthcoming paper. The problems solved in this paper are (3) on a unit square. Problems with constant and variable α and β are examined, both standard Nedelec interpolation and the operator-based interpolations described in the previous section are used, and the smoothers described in the previous section are employed. (Note that operator-based interpolation produces multilevel structures that do not satisfy exactness.) $V(1,1)$ cycles are used, and all experiments are performed with a zero righthand side, a random initial, and a stopping criterion of twelve order magnitude reduction of the $l2-$norm of the initial residual. The schemes used are

- Method 1: Hybrid Gauss-Seidel smoother- all nodes updated then all edges.
- Method 2: Intermediate Gauss-Seidel smoother- pointwise update one node and then all edges branching out of it in turn.
- Method 3: Node-edge decoupling on coarse grids- i.e., only the block-diagonal terms of (7) are used on the coarse grids (see the discussion after (7)).
- Method 4: Multiplicative Schwarz smoother- simultaneously solve all edges connected to a node. The nodal degrees of freedom are not involved; this method is used for comparison.

Problem 1: $\alpha = 1$, $\beta =$ constant. This experiment was conducted to illustrate the robustness of the scheme as $\beta \to 0$, so that the operator becomes singular. Table 1 contains the $V(1,1)$ iteration counts. For each iteration column, the left count is for Nedelec interpolation and the right count for operator-based interpolation. From these results, all methods are robust with respect to the size of β. In particular, not preserving exactness on coarse grids does not affect the convergence. Also, the number of iterations decreases as the smoother becomes "stronger," and Method 3 performs just as well as Method 1 because $\nabla\beta = 0$ so that the edge and nodal components of the divergence-free near-nullspace of (2) decouple.

Problem 2: Variable α, β. We consider the performance of this multigrid method on variable coefficient problems. Problem 2a will involve coefficients

$$\begin{aligned} \alpha &= (2+\sin 20\pi x)^2(2+\cos 20\pi y)^2 \\ \beta &= (2+\cos 20\pi x)^2(2+\sin 20\pi y)^2, \end{aligned}$$

and Problem 2b will involve grid-aligned jump coefficients

$$\begin{aligned} \alpha &= C(2+\sin 20\pi x)^2(2+\cos 20\pi y)^2 \\ \beta &= C(2+\cos 20\pi x)^2(2+\sin 20\pi y)^2, \end{aligned}$$

where jump C is described in Fig. 3. The results are tabulated in Table 2. For both of these problems, we see poor performance when Nedelec interpolation is used, and a dramatic improvement when operator-based interpolation is used. Note that Method 1 performs better than Method 3, as is expected because $\|\nabla\beta\|$ is large. Also note that Method 2 performs better than Method 4. Both of these methods do not satisfy exactness on the coarse grids, but only Method 2 has systems for

0.1	10^2
10	10^4

Figure 3: Jump C of Problem 2b.

the curl-free component on the coarse grids. Lastly, note that the performance of all methods with operator-based interpolation are unaffected by jumps in the coefficients, whereas coefficient jumps noticeably affects the convergence when Nedelec interpolation is used.

Problem 3: Constant coefficient jumps. Our final experiment further examines jump coefficients, but now with jumps that are not grid-aligned on the coarse grids. The log scale of the coefficients are shown in Fig. 4. The order of magnitude of the jumps are around 10. Results are given in

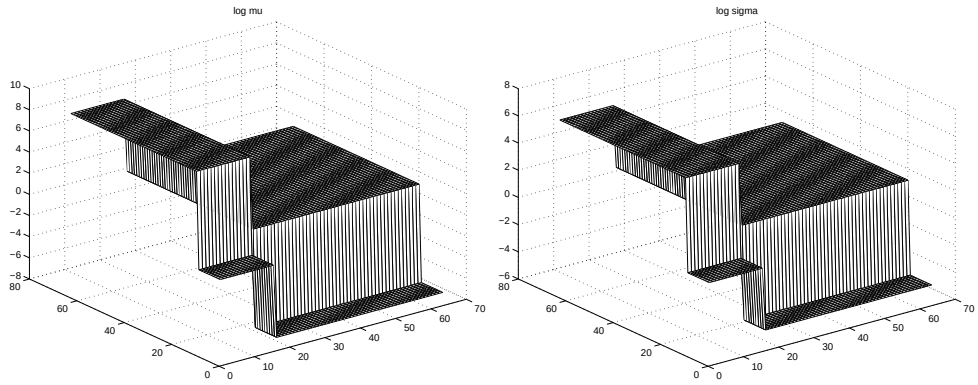

Figure 4: Log scale of jumps in μ and σ for Problem 3.

Table 3. Again operator-based interpolation performs better than Nedelec interpolation but not as dramatically this time. This can be accredited to the localization of the coefficient variations.

6 Conclusions

We presented a multigrid method for solving variable coefficient Maxwell's equations. This method does not construct interpolation operators based on multilevel commutativity constraints, and hence, does not satisfy exactness on the coarse levels. Rather, interpolation operators are separately developed for the nodes and edges, and operator-based techniques are used to capture the coefficient variations. Moreover, the relevant effects of exactness are obtained by having equations for the curl-free gradients on the coarse levels. Numerical results demonstrate the effectiveness of this algorithm.

References

[1] D. N. Arnold, *Differential complexes and numerical stability*, manuscript, 2003.

[2] D. N. Arnold, R. S. Falk, and R. Winther, *Multigrid in H(div) and H(curl)*, Numer. Math., 85 (2000), 197-218.

[3] D. Baldomir, *Geometry of Electromagnetic Systems*, Oxford University Press, New York, 1996.

[4] R. L. Bishop and S. I. Goldberg, *Tensor Analysis on Manifolds*, Dover Publication, New York, 1980.

[5] P. B. Bochev, C. J. Garasi, J. J. Hu, A. C. Robinson, R. S. Tuminaro, *An improved algebraic multigrid method for solving Maxwell's equations*, SIAM J. Sci. Comp., to appear.

[6] D. Boffi, *Fortin operator and discrete compactness for edge elements*, Numer. Math., 87 (2000), 229-246.

[7] A. Bossavit, *Computational Electromagnetism: Variational Formulations, Complementarity, Edge Elements*, Academic Press, San Diego, 1998.

[8] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer, New York, 1994.

[9] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, J. W. Ruge, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comp., 22 (2000), 1570-1592.

[10] J. E. Dendy, Jr., *Black box multigrid*, J. Comput. Phys., 48 (1982), 366-386.

[11] J. E. Dendy, Jr., *Black box multigrid for systems*, Appl. Math Comp., 19 (1986), 57-74.

[12] J. E. Dendy, Jr., *Semicoarsening multigrid for systems*, ETNA, 6 (1997), 97-105.

[13] H. Flanders, *Differential Forms with Applications to the Physical Sciences*, Dover Publication, New York, 1989.

[14] R. Hiptmair, *Canonical construction of finite elements*, Math. Comp., 68 (1999), 1325-1346.

[15] R. Hiptmair, *Multigrid method for Maxwell's equations*, SIAM J. Numer. Anal., 36 (1999), 204-225.

[16] R. Hiptmair, *Multilevel gauging for edge elements*, Comput., 64 (2000), 97-122.

[17] R. Hiptmair, *Finite elements in computational electromagnetism*, Acta Numerica, 11 (2002), 237-340.

[18] J. Jin, *The Finite Element Method in Electromagnetics*, John Wiley, New York, 2002.

[19] S. Reitzinger and J. Schoberl, *An algebraic multigrid method for finite element discretizations with edge elements*, Num. Lin. Alg. Appl., 9 (2002), 223-238.

[20] P. S. Vassilevski and J. Wang, *Multilevel iterative methods for mixed finite element discretizations of elliptic problems*, Numer. Math., 63 (1992), 503-520.

[21] H. Whitney, *Integration Theory*, Princeton University Press, 1957.

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

β	h	Method 1	Method 2	Method 3	Method 4
10^3	1/8	6/6	8/8	10/10	7/8
	1/16	12/11	8/8	13/13	7/7
	1/32	19/19	12/12	20/19	10/11
	1/64	24/24	14/15	24/24	12/13
10^2	1/8	12/12	8/9	15/15	8/7
	1/16	17/17	11/12	18/19	10/10
	1/32	23/23	15/15	23/23	12/13
	1/64	26/26	15/16	26/26	13/13
10^1	1/8	13/13	11/11	15/16	9/8
	1/16	18/17	12/13	19/19	11/11
	1/32	23/24	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
10^0	1/8	13/14	11/11	15/16	9/9
	1/16	17/18	13/12	19/18	12/11
	1/32	23/24	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
10^{-1}	1/8	13/13	11/11	15/15	8/9
	1/16	17/18	12/13	17/17	12/11
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
10^{-2}	1/8	13/13	11/11	13/13	8/9
	1/16	18/17	13/13	16/17	11/12
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
10^{-3}	1/8	14/13	11/11	13/13	9/9
	1/16	17/17	13/13	17/18	11/11
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	16/16	26/26	13/13
10^{-4}	1/8	13/13	11/11	13/13	8/8
	1/16	17/17	13/12	18/17	11/11
	1/32	23/23	15/15	23/23	13/13
	1/64	26/26	15/16	26/26	13/13
10^{-5}	1/8	13/13	11/11	13/13	8/9
	1/16	17/16	13/13	17/17	11/11
	1/32	23/23	15/15	24/23	13/13
	1/64	26/26	16/16	26/26	13/13

Table 1: V(1,1)-cycle results for Problem 1, $\alpha = 1$, $\beta =$ constant. The iteration counts on the left and right are for Nedelec interpolation and operator-based interpolation, respectively.

Problem	h	Method 1	Method 2	Method 3	Method 4
2a	1/8	41/12	38/11	41/15	37/16
	1/16	60/15	56/12	61/19	53/22
	1/32	70/18	60/12	70/24	56/22
	1/64	60/22	56/14	59/22	54/19
2b	1/8	62/10	62/9	62/13	48/11
	1/16	81/12	72/10	81/17	66/21
	1/32	> 100/15	93/11	> 100/18	89/18
	1/64	93/20	89/15	86/20	85/18

Table 2: V(1,1)-cycle results for variable coefficients, Problems 2a & 2b.

h	Method 1	Method 2	Method 4
1/8	43/15	35/15	26/14
1/16	34/15	30/12	22/11
1/32	27/18	24/14	18/12
1/64	26/18	23/14	17/12

Table 3: V(1,1)-cycle results for constant coefficient jumps.